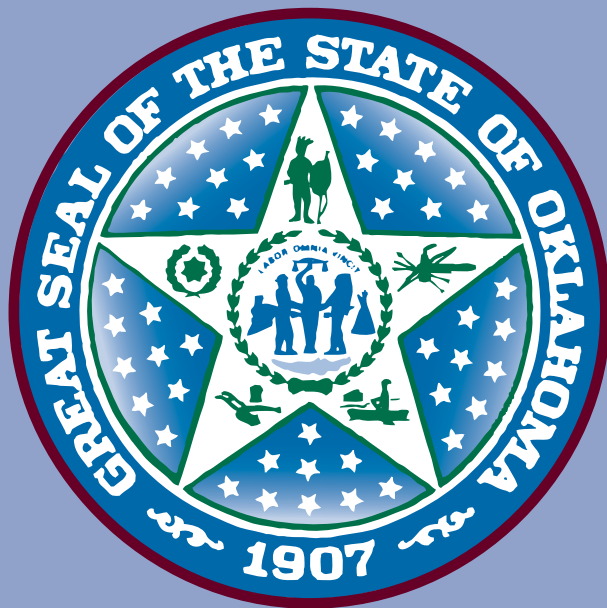


# Certification Examinations for Oklahoma Educators™

## Oklahoma Subject Area Tests™

# STUDY GUIDE

081 Computer Science



Oklahoma Commission  
for Teacher Preparation

OK-SG-FLD081-01

# TABLE OF CONTENTS

## STUDY GUIDE INTRODUCTION

PURPOSE OF THIS STUDY GUIDE.....	1-1
KEY FEATURES OF THIS STUDY GUIDE .....	1-1
FOR OTHER CEOE STUDY GUIDES.....	1-1
FOR FURTHER INFORMATION.....	1-1

## GENERAL INFORMATION ABOUT THE CERTIFICATION EXAMINATIONS FOR OKLAHOMA EDUCATORS

AN OVERVIEW OF THE TESTING PROGRAM.....	1-2
Test Development Process .....	1-2
Characteristics of the CEOE Tests.....	1-3
Test Descriptions .....	1-3
Test Administration.....	1-8
Score Reports.....	1-8
HOW TO PREPARE FOR THE TEST .....	1-8
Study the Competencies .....	1-8
Identify Resources .....	1-9
Study Techniques .....	1-9
Review the Practice Test Questions .....	1-9
SAMPLE TEST DIRECTIONS .....	1-10
Sample General Test Directions.....	1-10
Sample Constructed-Response Assignment Directions .....	1-11
THE DAY OF THE TEST.....	1-12
Preparation.....	1-12
At the Test Site.....	1-12

## FIELD-SPECIFIC INFORMATION

INTRODUCTION.....	2-1
Test Competencies .....	2-1
Practice Test Questions .....	2-1
TEST COMPETENCIES .....	2-2
PRACTICE TEST QUESTIONS AND ANSWERS .....	2-7
Practice Selected-Response Questions .....	2-7
Practice Constructed-Response Assignment.....	2-18
A Very Good Response to the Practice Constructed-Response Assignment.....	2-19
CONSTRUCTED-RESPONSE ASSIGNMENT SCORING.....	2-21
Sample Performance Characteristics for Constructed-Response Assignments .....	2-21
Sample Scoring Scale for Constructed-Response Assignments .....	2-21

*Readers should be advised that this study guide, including many of the excerpts used herein, is protected by federal copyright law.*

Copyright © 2009 by the Oklahoma Commission for Teacher Preparation and Pearson Education, Inc. or its affiliate(s). All rights reserved.  
Evaluation Systems, Pearson, P.O. Box 226, Amherst, MA 01004

Certification Examinations for Oklahoma Educators, CEOE, Oklahoma General Education Test, OGET, Oklahoma Professional Teaching Examination, OPTE, Oklahoma Subject Area Tests, and OSAT are trademarks, in the U.S. and/or other countries, of the Oklahoma Commission for Teacher Preparation and Pearson Education, Inc. or its affiliate(s).

Pearson and its logo are trademarks in the U.S. and/or other countries of Pearson Education, Inc. or its affiliate(s).



# STUDY GUIDE INTRODUCTION AND GENERAL INFORMATION ABOUT THE CERTIFICATION EXAMINATIONS FOR OKLAHOMA EDUCATORS

---

The first two sections of the study guide are available in a separate PDF file. Click the link below to view or print these sections.

[Study Guide Introduction and General Information About the Certification Examinations for Oklahoma Educators](#)





# FIELD-SPECIFIC INFORMATION

- Test Competencies
  - Practice Test Questions and Answers
  - Constructed-Response Assignment Scoring
- 

## INTRODUCTION

This section includes a list of the test competencies, as well as a set of practice selected-response (multiple-choice) questions and one or more practice constructed-response assignments (if applicable), for the test field included in this study guide.

### Test Competencies

The test competencies are broad, conceptual statements that reflect the subject-matter skills, knowledge, and understanding an entry-level educator needs to teach effectively in Oklahoma public schools. The list of test competencies for each test field represents the **only** source of information about what a specific test will cover and therefore should be reviewed carefully.

The descriptive statements that follow the competencies are included to provide examples of possible content covered by each competency. These descriptive statements are neither exhaustive nor exclusionary.

### Practice Test Questions

The practice selected-response questions and any practice constructed-response assignments included in this section are designed to give you an introduction to the nature of the questions included in this OSAT test field. The practice test questions represent the various types of questions you may expect to see on an actual test; however, they are **not** designed to provide diagnostic information to help you identify specific areas of individual strengths and weaknesses or to predict your performance on the test as a whole.

To help you prepare for your OSAT, each practice selected-response question in this section is preceded by the competency it measures and followed by a brief explanation of the correct response. On the actual test, the competencies, correct responses, and explanations will **not** be given.

If the test field included in this guide has a constructed-response assignment, a sample response is provided immediately following the practice constructed-response assignment. The sample response in this guide is for illustrative purposes only. Your written response should be your original work, written in your own words, and not copied or paraphrased from some other work.

A description of the process that is used for scoring the constructed-response assignment is provided in addition to the OSAT performance characteristics and score scale.

When you are finished with the practice test questions, you may wish to go back and review the entire list of test competencies and descriptive statements for your test field.

## TEST COMPETENCIES: COMPUTER SCIENCE

### SUBAREAS:

- I. Computer Use in Educational Environments
- II. Computer System Concepts
- III. Program Design and Algorithms
- IV. Programming and Program Testing

### SUBAREA I—COMPUTER USE IN EDUCATIONAL ENVIRONMENTS

#### Competency 0001

**Understand basic concepts related to the operation and use of computers and technology in educational environments.**

*The following topics are examples of content that may be covered under this competency.*

Demonstrate knowledge of procedures for installing and using peripherals and other devices and for troubleshooting hardware and software problems.

Demonstrate knowledge of concepts and terminology related to telecommunications.

Recognize types, characteristics, and uses of telecommunications tools and resources, including video conferencing and distance learning.

Demonstrate familiarity with equity issues regarding computer use (e.g., students with special needs, students with limited English proficiency).

Demonstrate familiarity with issues related to the privacy of electronic student information.

#### Competency 0002

**Understand the characteristics and uses of productivity software.**

*The following topics are examples of content that may be covered under this competency.*

Demonstrate knowledge of the features and uses of word-processing and desktop-publishing software.

Demonstrate knowledge of how to use spreadsheets to organize, analyze, and display data and how to integrate spreadsheet data into word-processing documents.

Demonstrate knowledge of how to design and manipulate databases, including relational databases, to organize data and create custom reports.

Demonstrate knowledge of terminology and concepts related to video and digital images (e.g., resolution, file formats, compression).

Demonstrate knowledge of features and uses of video editing and graphic design software.

Demonstrate familiarity with Web-page creation tools and concepts related to publishing material for the Web.

## **SUBAREA II—COMPUTER SYSTEM CONCEPTS**

### **Competency 0003**

**Understand basic terminology related to computer architecture and characteristics of computer architecture.**

*The following topics are examples of content that may be covered under this competency.*

Demonstrate knowledge of terminology related to computer architecture.

Identify characteristics and functions of computer components (e.g., CPU, RAM, secondary memory).

Demonstrate knowledge of data representation at the machine level (e.g., floating point, integer, character).

Demonstrate knowledge of data storage and how data is transferred from one location to another (e.g., registers, memory hierarchy).

Recognize the steps in the machine cycle and their synchronization.

Translate between binary, decimal, and hexadecimal number systems.

### **Competency 0004**

**Understand characteristics and functions of operating systems.**

*The following topics are examples of content that may be covered under this competency.*

Recognize the roles and functions of a computer's operating system.

Compare the characteristics and functions of single-user and multiuser systems.

Recognize the characteristics and functions of computer components and processes (e.g., devices, virtual memory, files, multitasking) and how the operating system manages them.

Recognize characteristics and functions of utility programs.

### **Competency 0005**

**Understand types and characteristics of computer networks.**

*The following topics are examples of content that may be covered under this competency.*

Demonstrate knowledge of terminology and concepts related to computer networks.

Demonstrate knowledge of various network configurations (e.g., peer to peer, client server) and their characteristics and the functions and characteristics of hubs, routers, and switches.

Recognize characteristics of local area networks (LANs), wide area networks (WANs), and wireless configurations.

Identify the basic structure and features of the Internet.

Demonstrate knowledge of network protocols and concepts of data transfer on the Internet.

Demonstrate knowledge of security issues related to networks and the Internet (e.g., firewalls, data encryption, malware, phishing).

## Competency 0006

**Understand the interactions between people and information systems and the social aspects of computing.**

*The following topics are examples of content that may be covered under this competency.*

Recognize the roles, responsibilities, and levels of access of various individuals who interact with information systems (e.g., programmer, Web site administrator, database administrator, end user).

Analyze issues related to the legal and ethical use of computer technology, including security policies.

Demonstrate knowledge of issues related to privacy and intellectual property rights when dealing with electronic data and information.

Analyze significant historical events and trends related to computing.

## SUBAREA III—PROGRAM DESIGN AND ALGORITHMS

### Competency 0007

**Understand principles and procedures for designing a program.**

*The following topics are examples of content that may be covered under this competency.*

Apply principles and concepts related to object-oriented programming.

Demonstrate knowledge of the steps in the process of program design.

Analyze flowcharts, schematic drawings, and pseudocode.

Demonstrate knowledge of characteristics and uses of top-down, bottom-up, and object-oriented design methodologies.

Apply principles of user interface design.

### Competency 0008

**Understand concepts and principles of modularization and data encapsulation in computer programming.**

*The following topics are examples of content that may be covered under this competency.*

Demonstrate knowledge of characteristics of program modules (e.g., functions, objects) and modularization strategies.

Apply principles of inheritance, polymorphism, and abstraction in program design.

Demonstrate knowledge of data encapsulation and its role in maintaining data integrity.

Recognize characteristics and uses of libraries and predefined classes.

Demonstrate knowledge of function calls, parameters, and parameter-passing techniques.

Apply knowledge of the use of constructors in creating objects.

Recognize characteristics of event-driven programming (e.g., input and output procedures, error handling).

## Competency 0009

### **Understand characteristics and uses of algorithms in high-level languages.**

*The following topics are examples of content that may be covered under this competency.*

Understand the general characteristics and the role of algorithms in computing.

Apply knowledge of the characteristics and uses of search algorithms (e.g., linear, binary).

Demonstrate knowledge of the characteristics and uses of sorting algorithms.

Determine algorithm output.

Analyze various types of algorithms (e.g., time-and-space trade-offs, big-O notation).

## Competency 0010

### **Understand principles and procedures for program development and implementation.**

*The following topics are examples of content that may be covered under this competency.*

Demonstrate knowledge of the steps in the programming process.

Apply knowledge of how to develop robust programs with effective user interfaces.

Recognize the purposes of programming style conventions (e.g., indenting, spacing, comments) and their appropriate application.

Apply knowledge of object-oriented development strategies.

Apply knowledge of strategies for modifying existing programs.

Demonstrate knowledge of software tools for developing programs.

Understand steps in the process of program execution (e.g., translation, linking, loading) in platform-dependent and platform-independent programming languages.

## SUBAREA IV—PROGRAMMING AND PROGRAM TESTING

### Competency 0011

#### **Understand types and characteristics of programming languages.**

*The following topics are examples of content that may be covered under this competency.*

Recognize differences in levels of programming language (e.g., machine, assembly, high-level).

Demonstrate knowledge of the characteristics, uses, strengths, and limitations of various types of high-level languages.

Recognize characteristics and functions of compilers and interpreters.

Demonstrate knowledge of the characteristics of various programming paradigms (e.g., imperative, functional, object-oriented).

Demonstrate knowledge of the historical development of programming languages.

## Competency 0012

**Understand types and characteristics of statements, operators, and control structures in high-level languages.**

*The following topics are examples of content that may be covered under this competency.*

Recognize the characteristics and uses of operators.

Recognize the characteristics and uses of statements (e.g., assignment, input/output, declaration).

Demonstrate knowledge of the characteristics and uses of conditional control structures.

Demonstrate knowledge of the characteristics and uses of repetitive control structures.

## Competency 0013

**Understand characteristics and applications of data types, structures, and abstraction mechanisms in high-level languages.**

*The following topics are examples of content that may be covered under this competency.*

Recognize the properties and uses of data types (e.g., integer, character, Boolean).

Demonstrate knowledge of the characteristics and uses of constants, variables, classes, functions, and parameters.

Analyze the characteristics and uses of inheritance and classes in object-oriented programming.

Analyze the characteristics and uses of data structures (e.g., stacks, trees, arrays, heaps, linked lists).

## Competency 0014

**Understand basic concepts related to code analysis, program testing, and documentation.**

*The following topics are examples of content that may be covered under this competency.*

Demonstrate knowledge of program correctness issues and practices (e.g., testing program results, test data design).

Analyze code segments to identify errors and determine output.

Recognize common programming errors.

Apply procedures for locating errors in computer programs.

Recognize the characteristics and purposes of user and system documentation of programs.

Apply principles of appropriate program documentation.

Demonstrate knowledge of the use of comments.

## PRACTICE TEST QUESTIONS AND ANSWERS: COMPUTER SCIENCE

### Practice Selected-Response Questions

#### Competency 0002

Understand the characteristics and uses of productivity software.

---

1. A teacher is using layers to create an image file with a graphic design program. Using layers to create the image provides the teacher with the ability to:
  - A. import the image file into another graphics program.
  - B. save the layers in a file format that can be used in a Web page.
  - C. create images that use a greater number of colors.
  - D. edit the layers independently of each other.

**Correct Response: D.** In a graphic design program, layers provide the ability to create the elements of an image separately so that each element can be manipulated and modified without affecting the others. When the elements are complete, the layers can be merged to incorporate all of the elements into a single image.

### Competency 0003

Understand basic terminology related to computer architecture and characteristics of computer architecture.

---

2. Immediately after an arithmetic calculation is performed, the result of the calculation is stored in:
  - A. the arithmetic logic unit.
  - B. a data bus.
  - C. the main memory.
  - D. a register.

**Correct Response: D.** Registers serve as temporary holding places for data being manipulated by the CPU. Registers hold the inputs to the arithmetic logic unit's circuitry and provide storage space for the results produced by that unit.

### Competency 0004

Understand characteristics and functions of operating systems.

---

3. In a computer system, virtual memory is a technique that:
  - A. compresses files so that more memory is available to a running application.
  - B. allows available hard drive space to be used to supplement a computer's memory.
  - C. permits unused read-only memory to be used by an application for temporary data storage.
  - D. provides the ability to move information in and out of memory more quickly.

**Correct Response: B.** Virtual memory is a method of extending the apparent size of a computer's random access memory (RAM) by swapping pages of data between RAM and the hard disk as needed.

### Competency 0005

Understand types and characteristics of computer networks.

---

4. Which of the following is a characteristic of a client-server network?
- A. A host computer transfers data to client computers by means of a token of code.
  - B. A host computer can be used only for tasks that cannot be performed by a client computer.
  - C. Application files installed on a client computer can access data files stored on a host computer.
  - D. All computers on a network can perform the duties of both a host and a client.

**Correct Response: C.** In a client/server network, software applications installed on the client computers can access data files stored on the computer's hard drive or data files stored on the network server. Files stored on the server can be accessed by multiple network users.

**Competency 0007**

Understand principles and procedures for designing a program.

---

5. Which of the following is a defining characteristic of an overloaded method?
- A. A method takes more than one parameter.
  - B. A method returns a value of a specific type.
  - C. Methods with the same name take different numbers or types of parameters.
  - D. Two or more methods are defined and implemented within the same class.

**Correct Response: C.** Within a class, several methods may have the same name. A method can have any number of overloaded versions as long as their parameter lists differ in the number or types of parameters.

**Competency 0008**

Understand concepts and principles of modularization and data encapsulation in computer programming.

---

6. Which of the following actions best illustrates the principle of abstraction in object-oriented programming?
- A. assigning a value to a newly created object
  - B. identifying the entities in a process with which a program interacts
  - C. identifying the flow of control in a real-world system
  - D. passing parameters from a method to the caller

**Correct Response: B.** Abstraction refers to the principle of separating the description of program entities from their implementation. A program needs to know only an entity's name and description in order to interact with it; the program does not need to know an entity's implementation details.

**Competency 0010**

Understand principles and procedures for program development and implementation.

---

7. Which of the following is the best example of how UML is commonly used in computer programming?
- A. visualizing the components of an object-oriented software system
  - B. creating a template for a graphical user interface
  - C. checking for invalid input during program execution
  - D. writing pseudocode specific to object-oriented languages

**Correct Response: A.** UML is a visual language that includes a set of graphical notation techniques to create visual abstract models of object-oriented software systems.

## Competency 0011

Understand types and characteristics of programming languages.

---

8. A significant difference between compilers and interpreters is that a compiler:
- A. generates error messages.
  - B. creates an executable file.
  - C. performs lexical analysis on the code.
  - D. compares words in the program to a list of valid commands.

**Correct Response: B.** A compiler is a program that reads the human-readable statements of a computer program and translates the statements into machine-readable object code that can then be executed. An interpreter translates and runs each program statement line by line.

## Competency 0012

Understand types and characteristics of statements, operators, and control structures in high-level languages.

---

9. Use the code segment below to answer the question that follows.

```
switch (expression)
```

```
case (expA)
```

```
    x = 1  
    break;
```

```
case (expB)
```

```
    x = 2  
    break;
```

```
case (expC)
```

```
    x = 3  
    break;
```

```
default: x = 0
```

Which of the following sets of statements is equivalent to the code segment shown above?

- A. If (*expA*) then *x* = 1 endif  
If (*expB*) then *x* = 2 endif  
If (*expC*) then *x* = 3 endif  
*x* = 0
- B. If (*expA*) then *x* = 1 else *x* = 0 endif  
If (*expB*) then *x* = 2 else *x* = 0 endif  
If (*expC*) then *x* = 3 else *x* = 0 endif
- C. If (*expA*) then *x* = 1 else  
If (*expB*) then *x* = 2 else  
If (*expC*) then *x* = 3  
endif  
endif  
endif  
else *x* = 0
- D. If (*expA*) then *x* = 1 else  
If (*expB*) then *x* = 2 else  
If (*expC*) then *x* = 3 else *x* = 0  
endif  
endif  
endif

**Correct Response: D.** In the case statement, if **case (exp A)** is true,  $x$  is assigned the value of 1, execution of the case statement is finished, and **case (exp B)** and **case (exp C)** are not evaluated. If **case (exp A)** is not true, then the next case statement is evaluated. The same is true for **case (exp B)** and **case (exp C)**. If the case is true, execution of the case statement is finished; otherwise program execution proceeds to the next case. This series of steps can be represented by a set of nested If ... then ... else statements with the value of  $x$  being assigned the value of 0 within the final else statement.

### Competency 0013

Understand characteristics and applications of data types, structures, and abstraction mechanisms in high-level languages.

---

10. A programmer is writing an algorithm that needs to repeatedly access and remove the smallest number in a collection of numbers. Which of the following data structures will be most appropriate for the programmer to use to store the collection?
- A. heap
  - B. stack
  - C. linked list
  - D. array

**Correct Response: A.** A heap is a specialized tree-based data structure in which the root node always contains either the largest (max-heap) or smallest (min-heap) element in the list. Using a min-heap, the smallest element of the list is removed by removing the element at the root and reordering the remaining elements.

## Practice Constructed-Response Assignment

11. **Read the information below; then complete the exercise that follows.**

A computer programmer needs to write a program that will operate a bank's automatic teller machine (ATM). Each of the bank's customers has a checking and/or a savings account. All savings accounts earn interest on the entire balance, while only checking accounts with balances over \$1,000 earn interest on the entire balance. Checking accounts have a monthly service fee, while savings accounts do not. To access his or her accounts, a customer enters his or her personal identification number (PIN). Once the PIN has been validated, the customer can choose to make a deposit, make a withdrawal, or transfer funds between accounts. Customers can perform an unlimited number of transfers from checking to savings accounts, but the number of transfers from savings to checking is limited to five per month. The program needs to calculate the interest earned by the accounts and update the balances in each account as necessary. It also needs to update the cash balance of the ATM.

The program needs to store the following data:

- ATM cash balance (total amount of money in the ATM)
- Customer PIN and personal information
- Customer account balances
- Interest rates on the accounts
- Checking account service fee
- Number of transfers made from a customer's savings account to his or her checking account each month

Write a response that demonstrates your conceptual understanding of an object-oriented approach to solving this problem, using the principle of inheritance. In your response:

- identify the classes you would use in the program design;
- identify the data and functionality (as defined in the problem statement) that would be contained in each class; and
- describe how your program design demonstrates the principle of inheritance.

## A Very Good Response to the Practice Constructed-Response Assignment

The main classes that need to be developed for this application are ATM, Customer, Account, and CheckingAccount and SavingsAccount, which are derived from the class Account.

There would also be a data structure that keeps track of the list of customers.

### ATM

**Data** - keeps track of

- ATM cash balance

**Methods** - performs the following transactions

- Validate PIN
- Deposit funds
- Withdraw funds
- Transfer funds between accounts
- Update cash balance

Interacts with the Customer and Account classes and the data structure that holds the list of customers

### Customer

**Data** - keeps track of

- Customer's PIN
- Personal information
- List of accounts

Interacts with the CheckingAccount and the SavingsAccount classes

### Account

**Data** - keeps track of

- Account number
- The account balance
- The interest rate for the account

**Methods** - performs the following transactions

- Calculate the interest for the account
- Update balance

### CheckingAccount

(derived from Account)

**Data** - keeps track of

- Service fee

**Methods** - performs the following transactions

- Deduct service fee each month
- Calculate interest on account if balance is over \$1000 (overrides superclass method)

### SavingsAccount

(derived from Account)

**Data** - keeps track of

- Number of transfers from Savings to Checking each month

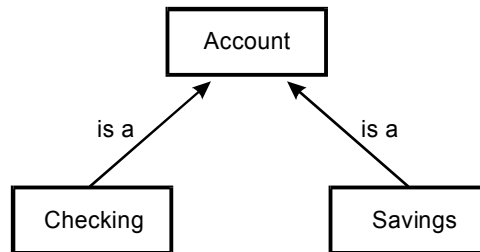
**Methods** - performs the following transactions

- Reset number of transfers to zero each month

(continued)

## A Very Good Response to the Practice Constructed-Response Assignment (continued)

This program design uses the principle of inheritance by defining a superclass `Account` with two subclasses, `SavingsAccount` and `CheckingAccount`. Both subclasses inherit the data members and functionality from `Account` and do not need to redefine them. In addition to the members inherited from `Account`, `CheckingAccount` would also include the service fee and would override the method in `Account` to calculate the interest on checking accounts with the appropriate balance. `SavingsAccount` needs to keep track of the number of transfers from `SavingsAccount` to `CheckingAccount`, since the number of transfers is limited to five per month.



## CONSTRUCTED-RESPONSE ASSIGNMENT SCORING

All responses to OSAT constructed-response assignments (written and oral) are scored using scoring scales that describe varying levels of performance. These scales were approved by committees of Oklahoma educators who reviewed both the performance characteristics and the scoring scales.

Each response is scored by multiple scorers according to standardized procedures during scoring sessions held immediately after each administration of the CEOE. Scorers with relevant professional backgrounds are oriented to these procedures before the scoring session and are carefully monitored during the scoring sessions.

A response to a constructed-response assignment is designated unscorable if it is blank, not on the assigned topic, illegible or unintelligible, not in the appropriate language, or of insufficient length to score. If you do not provide a scorable response for each constructed-response assignment on your test, you cannot pass the test regardless of your scores on the other section(s) of the test.

### Sample Performance Characteristics for Constructed-Response Assignments

<b>PURPOSE</b>	The extent to which the response achieves the purpose of the assignment
<b>SUBJECT MATTER KNOWLEDGE</b>	Accuracy and appropriateness in the application of subject matter knowledge
<b>SUPPORT</b>	Quality and relevance of supporting details
<b>RATIONALE</b>	Soundness of argument and degree of understanding of the subject matter

### Sample Scoring Scale for Constructed-Response Assignments

<b>SCORE POINT</b>	<b>SCORE POINT DESCRIPTION</b>
<b>4</b>	<p><b>The "4" response reflects a thorough knowledge and understanding of the subject matter.</b></p> <ul style="list-style-type: none"> <li>• The purpose of the assignment is fully achieved.</li> <li>• There is a substantial, accurate, and appropriate application of subject matter knowledge.</li> <li>• The supporting evidence is sound; there are high-quality, relevant examples.</li> <li>• The response reflects an ably reasoned, comprehensive understanding of the topic.</li> </ul>
<b>3</b>	<p><b>The "3" response reflects a general knowledge and understanding of the subject matter.</b></p> <ul style="list-style-type: none"> <li>• The purpose of the assignment is largely achieved.</li> <li>• There is a generally accurate and appropriate application of subject matter knowledge.</li> <li>• The supporting evidence generally supports the discussion; there are some relevant examples.</li> <li>• The response reflects a general understanding of the topic.</li> </ul>
<b>2</b>	<p><b>The "2" response reflects a partial knowledge and understanding of the subject matter.</b></p> <ul style="list-style-type: none"> <li>• The purpose of the assignment is partially achieved.</li> <li>• There is a limited, possibly inaccurate or inappropriate application of subject matter knowledge.</li> <li>• The supporting evidence is limited; there are few relevant examples.</li> <li>• The response reflects a limited, poorly reasoned understanding of the topic.</li> </ul>
<b>1</b>	<p><b>The "1" response reflects little or no knowledge and understanding of the subject matter.</b></p> <ul style="list-style-type: none"> <li>• The purpose of the assignment is not achieved.</li> <li>• There is little or no appropriate or accurate application of subject matter knowledge.</li> <li>• The supporting evidence, if present, is weak; there are few or no relevant examples.</li> <li>• The response reflects little or no reasoning about or understanding of the topic.</li> </ul>
<b>U</b>	<b>The response is unscorable because it is illegible, not written to the assigned topic, written in a language other than English, or of insufficient length to score.</b>
<b>B</b>	<b>There is no response to the assignment.</b>

**CERTIFICATION EXAMINATIONS FOR OKLAHOMA EDUCATORS  
EVALUATION SYSTEMS  
PEARSON  
AMHERST, MA 01004-9007**

[www.octp.org](http://www.octp.org)  
[www.ceoe.nesinc.com](http://www.ceoe.nesinc.com)

